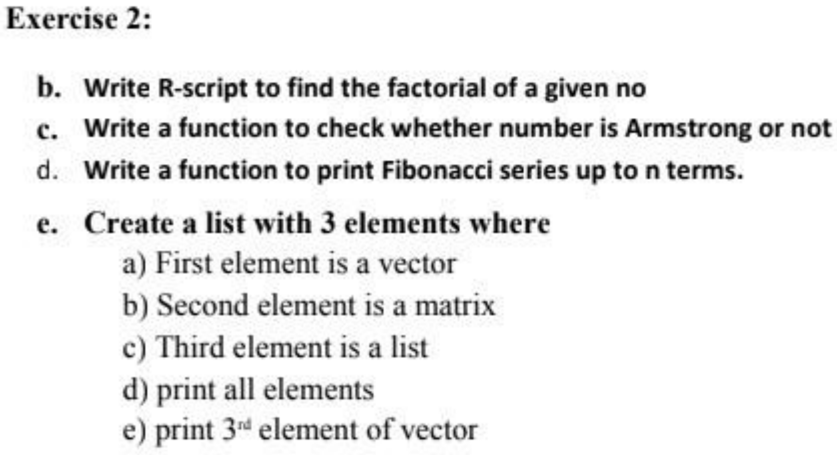
ASSIGNMENT-2

NAME:K.RAGHU
REG NO:22MIS7195



Exercise 2:

b. **Write R-script to find the factorial of a given no**

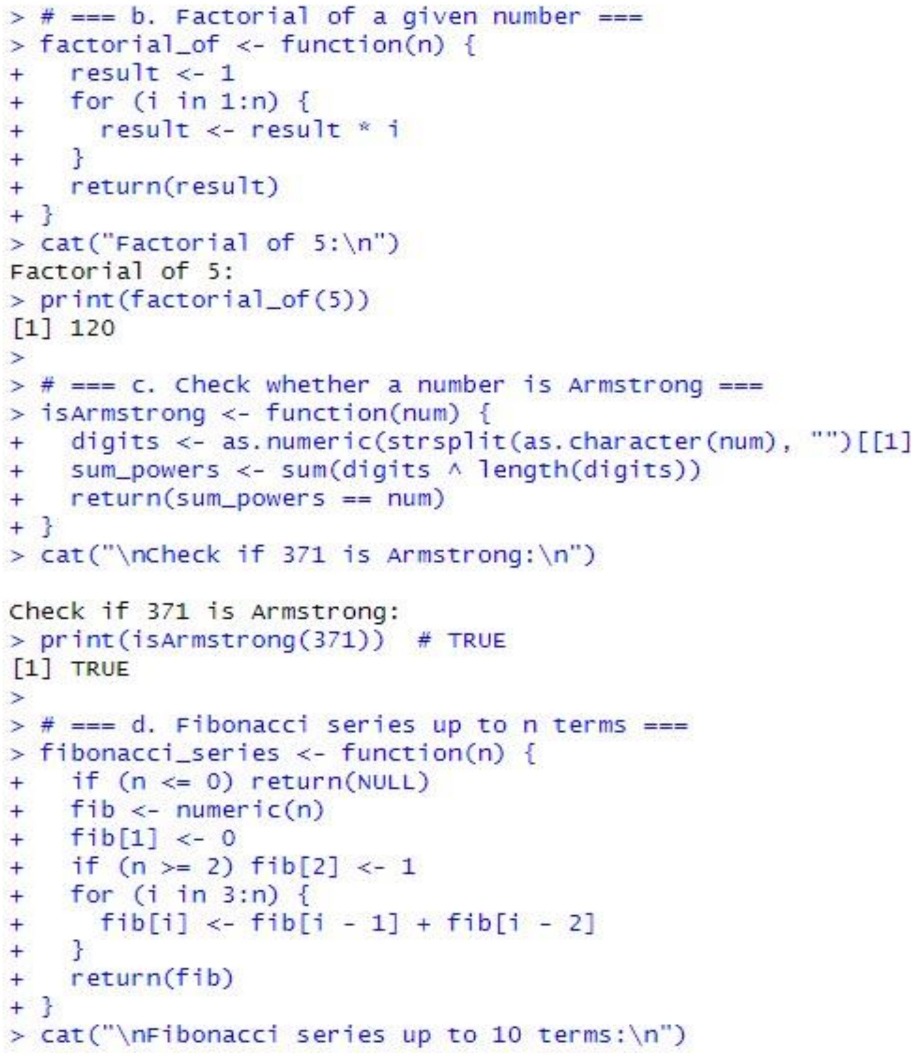
c. **Write a function to check whether number is Armstrong or not**

d. **Write a function to print Fibonacci series up to n terms.**

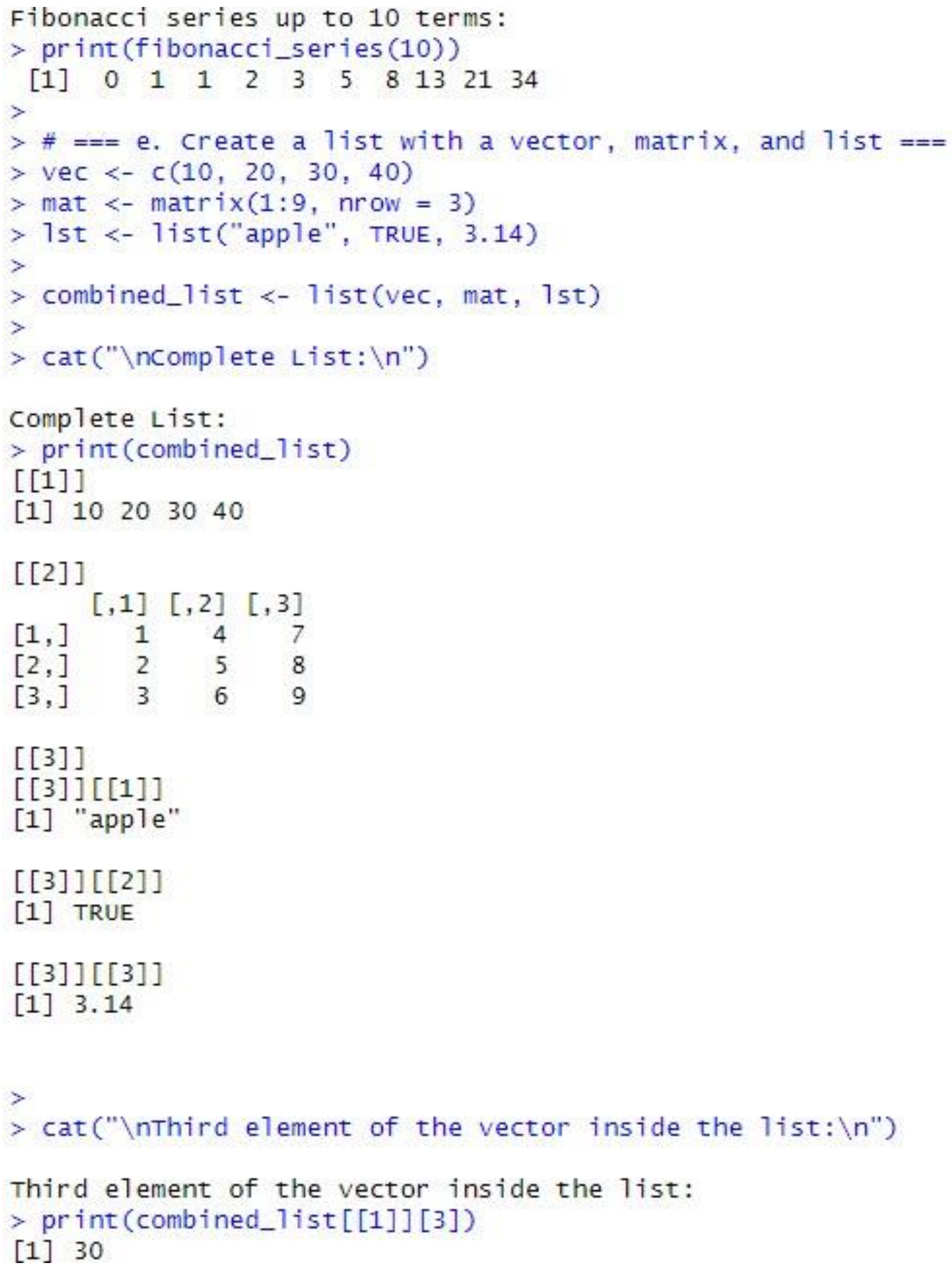
e. **Create a list with 3 elements where**

  a) First element is a vector
  b) Second element is a matrix
  c) Third element is a list
  d) print all elements
  e) print 3rd element of vector

OUTPUT:



```
> # === b. Factorial of a given number ===
> factorial_of <- function(n) {
+   result <- 1
+   for (i in 1:n) {
+     result <- result * i
+   }
+   return(result)
+ }
> cat("Factorial of 5:\n")
Factorial of 5:
> print(factorial_of(5))
[1] 120
>
> # === c. Check whether a number is Armstrong ===
> isArmstrong <- function(num) {
+   digits <- as.numeric(strsplit(as.character(num), "")[[1]
+   sum_powers <- sum(digits ^ length(digits))
+   return(sum_powers == num)
+ }
> cat("\nCheck if 371 is Armstrong:\n")

Check if 371 is Armstrong:
> print(isArmstrong(371))  # TRUE
[1] TRUE
>
> # === d. Fibonacci series up to n terms ===
> fibonacci_series <- function(n) {
+   if (n <= 0) return(NULL)
+   fib <- numeric(n)
+   fib[1] <- 0
+   if (n >= 2) fib[2] <- 1
+   for (i in 3:n) {
+     fib[i] <- fib[i - 1] + fib[i - 2]
+   }
+   return(fib)
+ }
> cat("\nFibonacci series up to 10 terms:\n")
```



```
Fibonacci series up to 10 terms:
> print(fibonacci_series(10))
 [1]  0  1  1  2  3  5  8 13 21 34
>
> # === e. Create a list with a vector, matrix, and list ===
> vec <- c(10, 20, 30, 40)
> mat <- matrix(1:9, nrow = 3)
> lst <- list("apple", TRUE, 3.14)
>
> combined_list <- list(vec, mat, lst)
>
> cat("\nComplete List:\n")

Complete List:
> print(combined_list)
[[1]]
[1] 10 20 30 40

[[2]]
     [,1] [,2] [,3]
[1,]    1    4    7
[2,]    2    5    8
[3,]    3    6    9

[[3]]
[[3]][[1]]
[1] "apple"

[[3]][[2]]
[1] TRUE

[[3]][[3]]
[1] 3.14


>
> cat("\nThird element of the vector inside the list:\n")

Third element of the vector inside the list:
> print(combined_list[[1]][3])
[1] 30
```

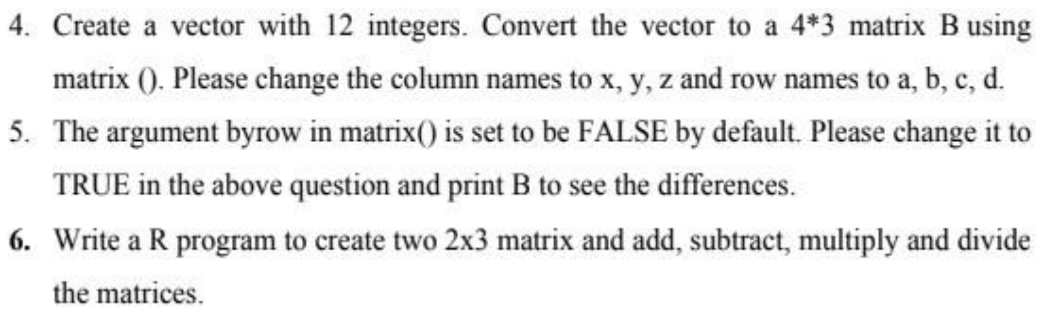
f. Matrix And Operations

Ex:

1. Write an R program to create a blank matrix.

   m = matrix (, nrow = 10, ncol = 5)

2. Write an R program to create a 3*2 matrix taking a given vector v= (2, 5, 8, 9, 7, 4) of numbers as input. Display the matrix.
3. Write a R program to create a 3*3 matrix taking a given vector A=(9,5,4,6,7,8,3,2,1) of numbers as input and define the column and row names. Display the matrix.

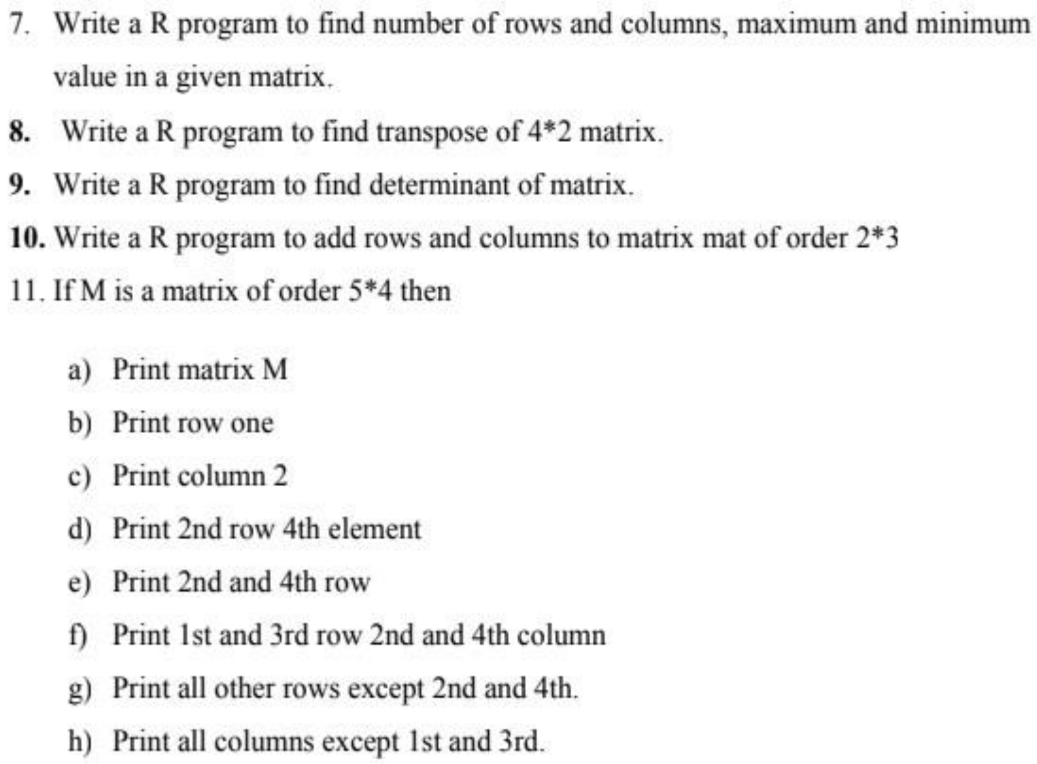


4. Create a vector with 12 integers. Convert the vector to a 4*3 matrix B using matrix (). Please change the column names to x, y, z and row names to a, b, c, d.

5. The argument byrow in matrix() is set to be FALSE by default. Please change it to TRUE in the above question and print B to see the differences.

6. Write a R program to create two 2x3 matrix and add, subtract, multiply and divide the matrices.

[redacted]



7. Write a R program to find number of rows and columns, maximum and minimum value in a given matrix.

8. Write a R program to find transpose of 4*2 matrix.

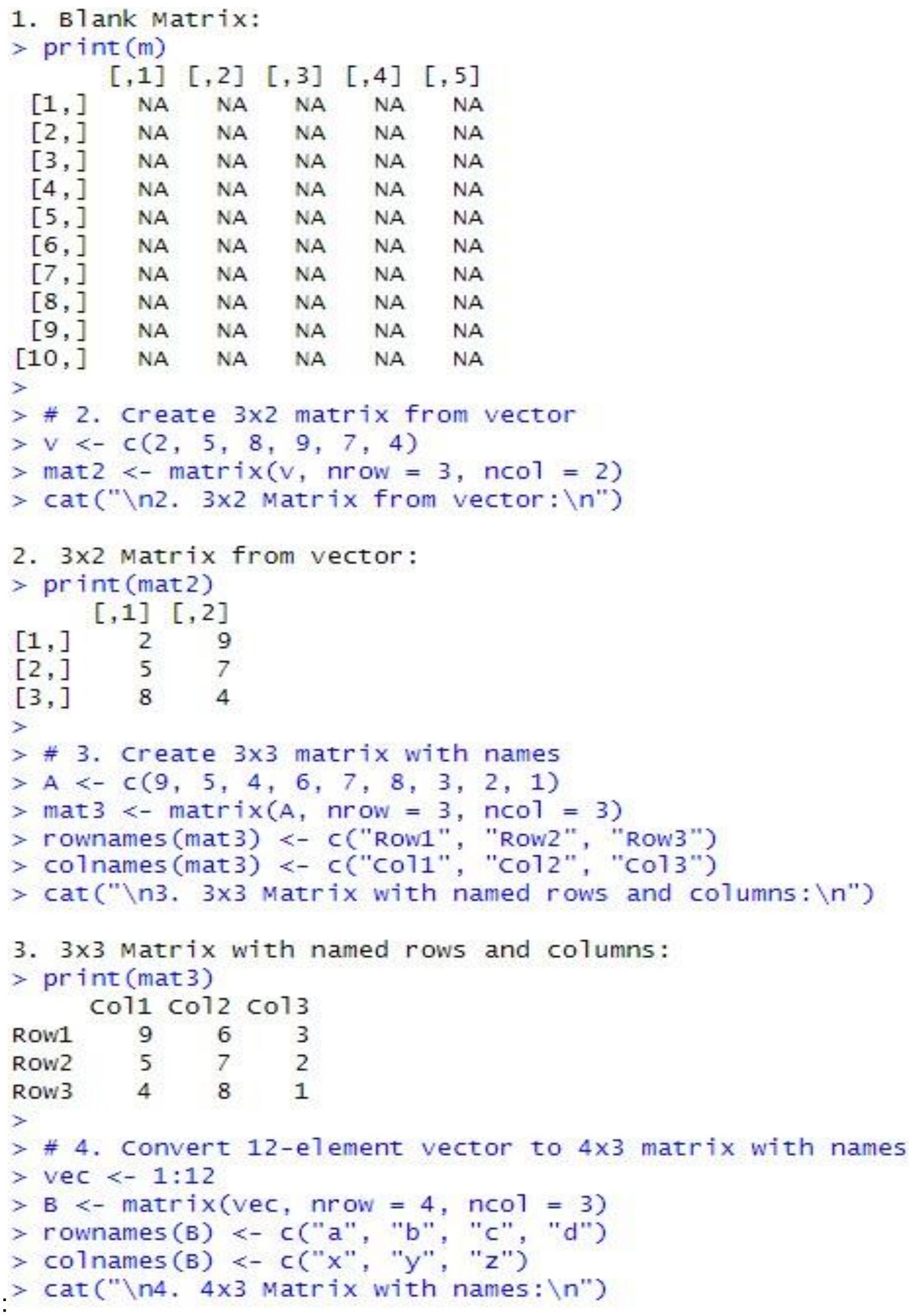
9. Write a R program to find determinant of matrix.

10. Write a R program to add rows and columns to matrix mat of order 2*3

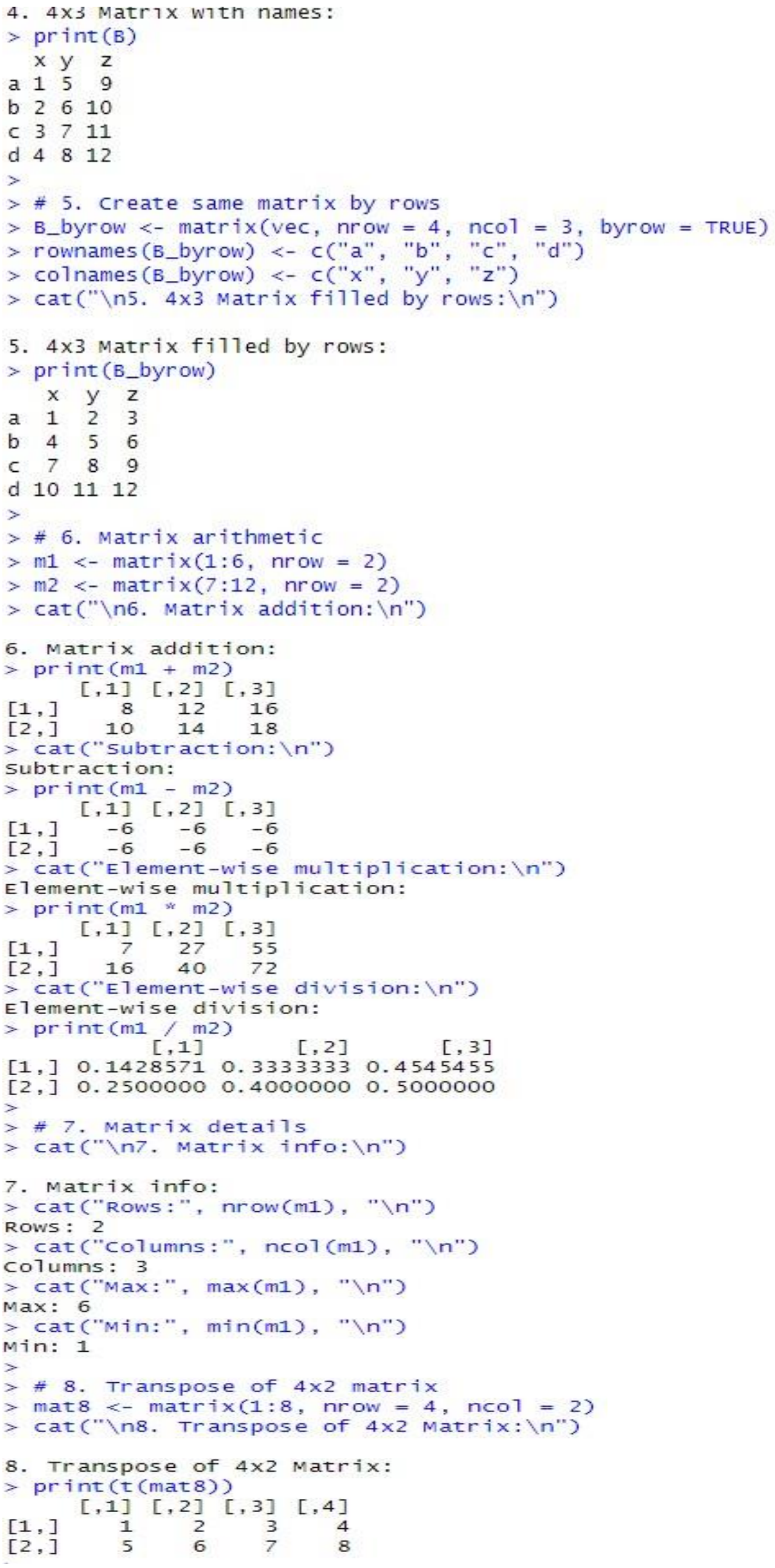
11. If M is a matrix of order 5*4 then

    a) Print matrix M
    b) Print row one
    c) Print column 2
    d) Print 2nd row 4th element
    e) Print 2nd and 4th row
    f) Print 1st and 3rd row 2nd and 4th column
    g) Print all other rows except 2nd and 4th.
    h) Print all columns except 1st and 3rd.

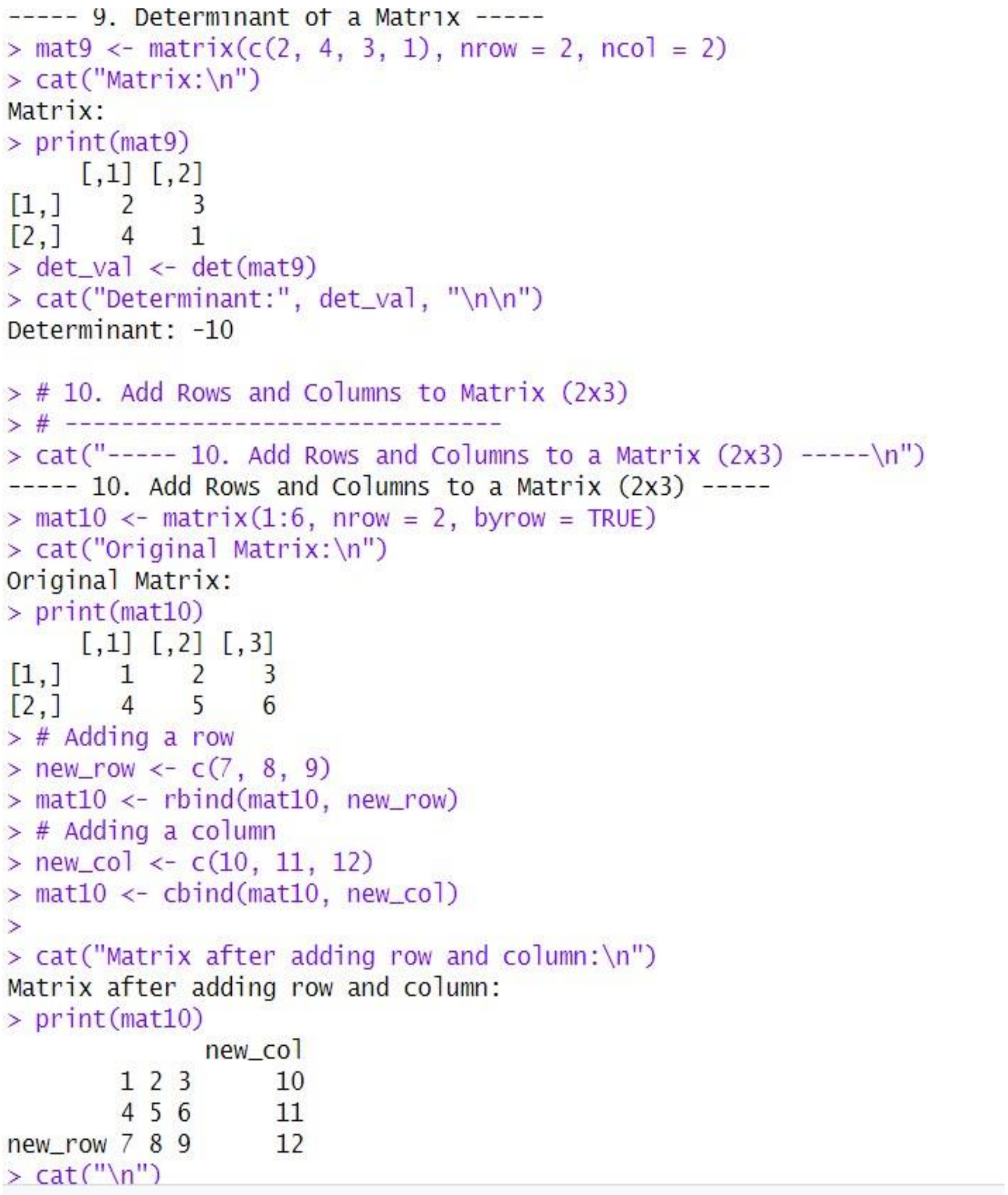
OUTPUT:



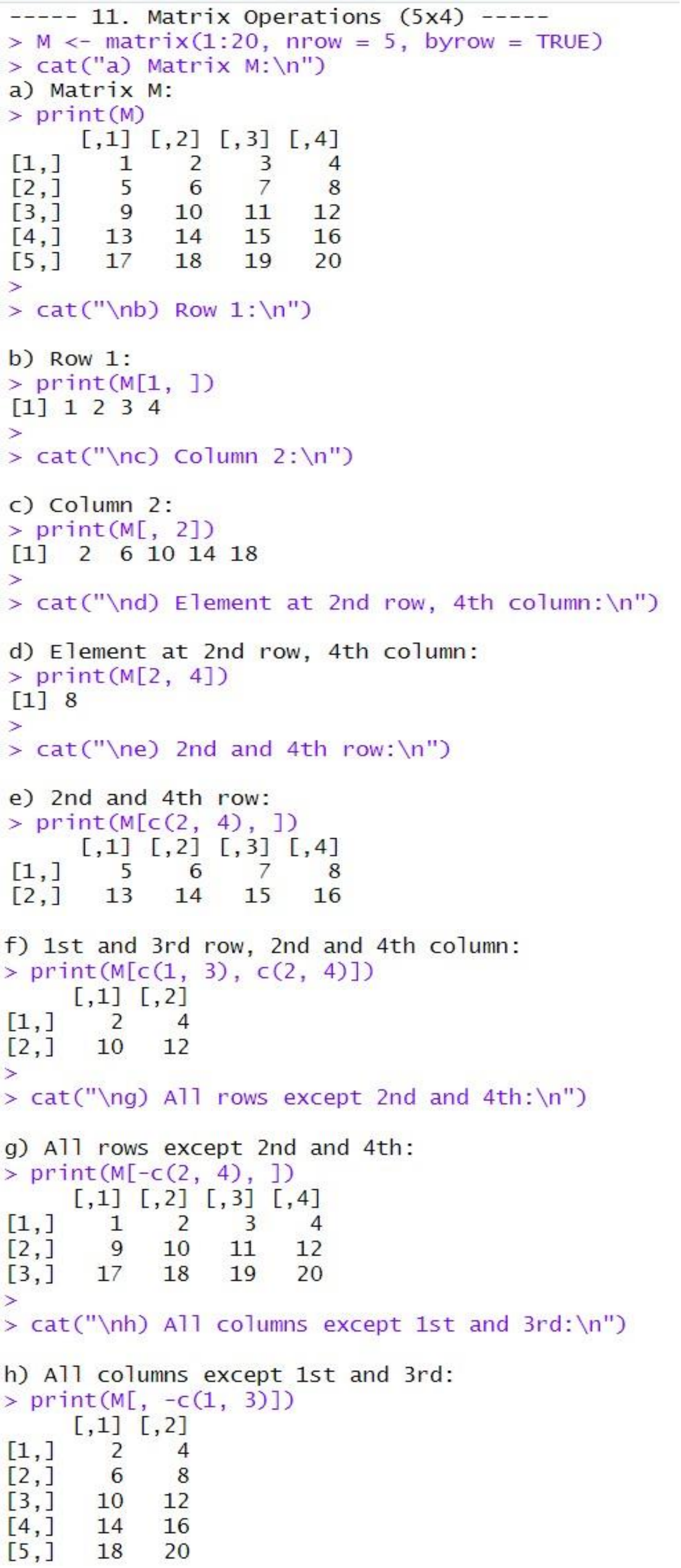
```
1. Blank Matrix:
> print(m)
      [,1] [,2] [,3] [,4] [,5]
 [1,]   NA   NA   NA   NA   NA
 [2,]   NA   NA   NA   NA   NA
 [3,]   NA   NA   NA   NA   NA
 [4,]   NA   NA   NA   NA   NA
 [5,]   NA   NA   NA   NA   NA
 [6,]   NA   NA   NA   NA   NA
 [7,]   NA   NA   NA   NA   NA
 [8,]   NA   NA   NA   NA   NA
 [9,]   NA   NA   NA   NA   NA
[10,]   NA   NA   NA   NA   NA
> 
> # 2. Create 3x2 matrix from vector
> v <- c(2, 5, 8, 9, 7, 4)
> mat2 <- matrix(v, nrow = 3, ncol = 2)
> cat("\n2. 3x2 Matrix from vector:\n")

2. 3x2 Matrix from vector:
> print(mat2)
     [,1] [,2]
[1,]    2    9
[2,]    5    7
[3,]    8    4
> 
> # 3. Create 3x3 matrix with names
> A <- c(9, 5, 4, 6, 7, 8, 3, 2, 1)
> mat3 <- matrix(A, nrow = 3, ncol = 3)
> rownames(mat3) <- c("Row1", "Row2", "Row3")
> colnames(mat3) <- c("Col1", "Col2", "Col3")
> cat("\n3. 3x3 Matrix with named rows and columns:\n")

3. 3x3 Matrix with named rows and columns:
> print(mat3)
     Col1 Col2 Col3
Row1    9    6    3
Row2    5    7    2
Row3    4    8    1
> 
> # 4. Convert 12-element vector to 4x3 matrix with names
> vec <- 1:12
> B <- matrix(vec, nrow = 4, ncol = 3)
> rownames(B) <- c("a", "b", "c", "d")
> colnames(B) <- c("x", "y", "z")
> cat("\n4. 4x3 Matrix with names:\n")
```



```
4. 4x3 Matrix with names:
> print(B)
  x y  z
a 1 5  9
b 2 6 10
c 3 7 11
d 4 8 12
>
> # 5. Create same matrix by rows
> B_byrow <- matrix(vec, nrow = 4, ncol = 3, byrow = TRUE)
> rownames(B_byrow) <- c("a", "b", "c", "d")
> colnames(B_byrow) <- c("x", "y", "z")
> cat("\n5. 4x3 Matrix filled by rows:\n")

5. 4x3 Matrix filled by rows:
> print(B_byrow)
   x  y  z
a  1  2  3
b  4  5  6
c  7  8  9
d 10 11 12
>
> # 6. Matrix arithmetic
> m1 <- matrix(1:6, nrow = 2)
> m2 <- matrix(7:12, nrow = 2)
> cat("\n6. Matrix addition:\n")

6. Matrix addition:
> print(m1 + m2)
     [,1] [,2] [,3]
[1,]    8   12   16
[2,]   10   14   18
> cat("Subtraction:\n")
Subtraction:
> print(m1 - m2)
     [,1] [,2] [,3]
[1,]   -6   -6   -6
[2,]   -6   -6   -6
> cat("Element-wise multiplication:\n")
Element-wise multiplication:
> print(m1 * m2)
     [,1] [,2] [,3]
[1,]    7   27   55
[2,]   16   40   72
> cat("Element-wise division:\n")
Element-wise division:
> print(m1 / m2)
          [,1]      [,2]      [,3]
[1,] 0.1428571 0.3333333 0.4545455
[2,] 0.2500000 0.4000000 0.5000000
>
> # 7. Matrix details
> cat("\n7. Matrix info:\n")

7. Matrix info:
> cat("Rows:", nrow(m1), "\n")
Rows: 2
> cat("Columns:", ncol(m1), "\n")
Columns: 3
> cat("Max:", max(m1), "\n")
Max: 6
> cat("Min:", min(m1), "\n")
Min: 1
>
> # 8. Transpose of 4x2 matrix
> mat8 <- matrix(1:8, nrow = 4, ncol = 2)
> cat("\n8. Transpose of 4x2 Matrix:\n")

8. Transpose of 4x2 Matrix:
> print(t(mat8))
     [,1] [,2] [,3] [,4]
[1,]    1    2    3    4
[2,]    5    6    7    8
```



```
----- 9. Determinant of a Matrix -----
> mat9 <- matrix(c(2, 4, 3, 1), nrow = 2, ncol = 2)
> cat("Matrix:\n")
Matrix:
> print(mat9)
     [,1] [,2]
[1,]    2    3
[2,]    4    1
> det_val <- det(mat9)
> cat("Determinant:", det_val, "\n\n")
Determinant: -10

> # 10. Add Rows and Columns to Matrix (2x3)
> # -------------------------------
> cat("----- 10. Add Rows and Columns to a Matrix (2x3) -----\n")
----- 10. Add Rows and Columns to a Matrix (2x3) -----
> mat10 <- matrix(1:6, nrow = 2, byrow = TRUE)
> cat("Original Matrix:\n")
Original Matrix:
> print(mat10)
     [,1] [,2] [,3]
[1,]    1    2    3
[2,]    4    5    6
> # Adding a row
> new_row <- c(7, 8, 9)
> mat10 <- rbind(mat10, new_row)
> # Adding a column
> new_col <- c(10, 11, 12)
> mat10 <- cbind(mat10, new_col)
>
> cat("Matrix after adding row and column:\n")
Matrix after adding row and column:
> print(mat10)
              new_col
        1 2 3      10
        4 5 6      11
new_row 7 8 9      12
> cat("\n")
```



```
----- 11. Matrix Operations (5x4) -----
> M <- matrix(1:20, nrow = 5, byrow = TRUE)
> cat("a) Matrix M:\n")
a) Matrix M:
> print(M)
     [,1] [,2] [,3] [,4]
[1,]    1    2    3    4
[2,]    5    6    7    8
[3,]    9   10   11   12
[4,]   13   14   15   16
[5,]   17   18   19   20
> 
> cat("\nb) Row 1:\n")

b) Row 1:
> print(M[1, ])
[1] 1 2 3 4
> 
> cat("\nc) Column 2:\n")

c) Column 2:
> print(M[, 2])
[1]  2  6 10 14 18
> 
> cat("\nd) Element at 2nd row, 4th column:\n")

d) Element at 2nd row, 4th column:
> print(M[2, 4])
[1] 8
> 
> cat("\ne) 2nd and 4th row:\n")

e) 2nd and 4th row:
> print(M[c(2, 4), ])
     [,1] [,2] [,3] [,4]
[1,]    5    6    7    8
[2,]   13   14   15   16

f) 1st and 3rd row, 2nd and 4th column:
> print(M[c(1, 3), c(2, 4)])
     [,1] [,2]
[1,]    2    4
[2,]   10   12
> 
> cat("\ng) All rows except 2nd and 4th:\n")

g) All rows except 2nd and 4th:
> print(M[-c(2, 4), ])
     [,1] [,2] [,3] [,4]
[1,]    1    2    3    4
[2,]    9   10   11   12
[3,]   17   18   19   20
> 
> cat("\nh) All columns except 1st and 3rd:\n")

h) All columns except 1st and 3rd:
> print(M[, -c(1, 3)])
     [,1] [,2]
[1,]    2    4
[2,]    6    8
[3,]   10   12
[4,]   14   16
[5,]   18   20
```